UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/054,307	01/22/2002	Maximiliano Gerardo Caceres	02929.000100	5426

5514 7590 12/07/2005

FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112

EXAMINER	
BAUM, RONALD	

ART UNIT	PAPER NUMBER
2136	

DATE MAILED: 12/07/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/054,307	CACERES ET AL.	
	Examiner	Art Unit	
	Ronald Baum	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *7/23/03,3/2/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1- 29 are pending for examination.

2. Claims 1- 29 are rejected.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 7,8 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 7,8 are directed towards "Computer code for...", whereas computer code or software is specifically non-statutory subject matter. The examiner assumes for the sake of applying art that the claims are directed to software embodied on a computer readable medium. Correction is required.

Claims 10-13 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 10-13 are directed towards "An agent for ...", whereas computer code or software or agents are specifically non-statutory subject matter. The examiner assumes for the sake of applying art that the claims are directed to software embodied on a computer readable medium. Correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 22 recites the limitation "... in *the second* target host ..." in reference to an existing claimed "... *second* target host ...". There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. Claims 1-29 are rejected under 35 U.S.C. 102(a) as being anticipated by Kargl, Frank, et al, "Protecting Web Servers from Distributed Denial of Service Attacks", Univ. of Ulm, 5/1/2001, entire document, http://www10.org/cdrom/papers/409/, ("Kargl").

4. As per claim 1; "A system for performing penetration testing of a target computer network by installing a remote agent in the target computer network, the system comprising:

a local agent

provided in a console and

configured to

receive and

execute commands [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack

on the targeted server(s). This involves the use of installing remote agents on the first network system and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects).];

a user interface

provided in the console and

configured to

send commands to and receive information from the local agent,

process the information, and

present the processed information [section 2.4-6];

a database

configured to store the information received from the local agent [section 2.2, 2.4-6];

a network interface

connected to the local agent and

configured to communicate via a network with

the remote agent installed in the target computer network [section 2.4-6];

and

security vulnerability exploitation modules for

execution by

the local agent and/or

the remote agent [section 2.4-6, whereas the installation of the zombie / daemons for the purpose of the subsequent denial of service attack clearly encompasses the security vulnerability exploitation; post module install/post install execution.].".

As per claim 4, this claim is the method for the apparatus/system claim 1 above, and is rejected for the same reasons provided for the claim 1 rejection, as such; "A method for performing penetration testing of a target computer network, comprising:

installing a remote agent in

the target computer network;

executing a command using

a local agent provided in a console;

receiving information from the local agent in

a user interface provided in the console;

presenting the information received from

the local agent to a user;

storing the information received from

the local agent in a database;

communicating via a network with

the remote agent installed in the target computer network; and

providing security vulnerability exploitation modules for

execution by

the local agent and/or

the remote agent.".

As per claim 7, this claim is the embodied software for the apparatus/system claim 1 above, and is rejected for the same reasons provided for the claim 1 rejection, as such; "Computer code for performing penetration testing of a target computer network, the computer code comprising code for:

installing a remote agent in

the target computer network;

executing a command using

a local agent provided in a console;

receiving information from the local agent in

a user interface provided in the console;

presenting the information received from

the local agent to a user;

storing the information received from

the local agent in a database;

communicating via a network with

the remote agent installed in the target computer network; and

providing security vulnerability exploitation modules for

execution by

the local agent and/or

the remote agent.".

5. Claim 2 ***additionally recites*** the limitation that; "The system of claim 1, wherein the user interface enables a user to

select one of the modules and

initiate execution of the selected module on either

the local agent or

the remote agent.".

The teachings of Kargl are directed towards such limitations (i.e., section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)).

As per claim 5, this claim is the method for the apparatus/system claim 2 above, and is rejected for the same reasons provided for the claim 2 rejection, as such; "The method of claim 4, further comprising:

selecting, using the user interface, one of the modules; and

initiating execution of the selected module on either

the local agent or

the remote agent.".

As per claim 8, this claim is the embodied software for the apparatus/system claim 2 above, and is rejected for the same reasons provided for the claim 2 rejection, as such; "The computer code of claim 7, further comprising code for:

selecting, using the user interface, one of the modules; and

initiating execution of the selected module on either

the local agent or

the remote agent.".

6. Claim 3 ***additionally recites*** the limitation that; "The system of claim 1, wherein

the user interface provides

a graphical representation of the target computer network.".

The teachings of Kargl are directed towards such limitations (i.e., section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select

possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface, where the section 2.4.1-2.4.5 are examples of Windows and 'X' (Unix) GUI's) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)).

As per claim 6, this claim is the method for the apparatus/system claim 3 above, and is rejected for the same reasons provided for the claim 3 rejection, as such; "The method of claim 4, further comprising

providing a graphical representation of the target computer network using the user interface.".

As per claim 9, this claim is the embodied software for the apparatus/system claim 3 above, and is rejected for the same reasons provided for the claim 3 rejection, as such; "The computer code of claim 7, further comprising

code for providing a graphical representation of the target computer network using the user interface.".

7. As per claim 10; "An agent for use in a system for performing penetration testing of a target computer network, the agent comprising:

a proxy server configured to

receive and execute system calls received via a network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system (i.e., routers and firewalls clearly encompass proxy servers, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)]; and

a virtual machine configured to

execute scripting language instructions received via the network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system (i.e., routers and firewalls clearly encompass proxy servers, and WEB based servers are JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and

subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)].".

8. Claim 11 ***additionally recites*** the limitation that; "The agent of claim 10, further comprising

an execution engine configured to

control

the proxy server and

the virtual machine,

wherein

the system calls and

the scripting language instructions

are routed to

the proxy server and

the virtual machine, respectively, by the execution engine.".

The teachings of Kargl are directed towards such limitations (i.e., section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted

server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)).

9. Claim 12 ***additionally recites*** the limitation that; "The agent of claim 11, further comprising

a remote procedure call module configured to

receive commands from the network formatted in

a remote procedure call protocol and

pass the commands to the execution engine.".

The teachings of Kargl are directed towards such limitations (i.e., section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which

inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., remote procedure call modules, distributed objects, daemons, CGI scripts, etc.)).

10. As per claim 13, this claim is the combination of claims 10-12 above, and is rejected for the same reasons provided for the claims 10-12 rejection, as such; "An agent for use in a system for performing penetration testing of a target computer network, the agent comprising:

a proxy server configured to

receive and execute system calls received via a network;

a virtual machine configured to

execute scripting language instructions received via the network;

a secure communication module configured to

provide secure communication between

the virtual machine and the network;

an execution engine configured to control

the proxy server and

the virtual machine,

wherein

the system calls and

the scripting language instructions

are routed to

the proxy server and

the virtual machine, respectively, by the execution engine;

a remote procedure call module configured to

receive commands via the network formatted in

a remote procedure call protocol and

pass the commands to the execution engine; and

a second secure communication module configured to

provide secure communication between

the remote procedure call module and the network.".

11. As per claim 14; "A method for performing penetration testing of a target network, comprising the steps of:

executing a first module in a console having a user interface,

the first module being configured to

exploit a security vulnerability in

a first target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the

attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects). The installation of the zombie / daemons for the purpose of the subsequent denial of service attack clearly encompasses the security vulnerability exploitation; post module install/post install execution.];

installing a first remote agent in the first target host,

the first remote agent being configured to

communicate with the console and a second remote agent [section 2.4-6];

and

executing a second module in the first remote agent,

the second module being configured to

exploit a security vulnerability in

a second target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the

attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents on the first network system (installed software modules, objects, etc.) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects). The installation of the zombie / daemons for the purpose of the subsequent denial of service attack clearly encompasses the security vulnerability exploitation; post module install/post install execution.].".

As per claim 16, this claim is the apparatus/system for the method claim 14 above, and is rejected for the same reasons provided for the claim 14 rejection, as such; "A system for performing penetration testing of a target network, comprising:

a console having a user interface;

a first module configured to

execute in the console to

exploit a security vulnerability in

a first target host of the target network;

a first remote agent installed in the first target host,

the first remote agent being configured to

communicate with the console and a second remote agent; and

a second module configured to

execute in the first remote agent to

exploit a security vulnerability in

a second target host of the target network.".

As per claim 18, this claim is the embodied software for the method claim 14 above, and is rejected for the same reasons provided for the claim 14 rejection, as such; "Computer code for performing penetration testing of a target network, the computer code comprising code for:

executing a first module in a console having a user interface,

the first module being configured to

exploit a security vulnerability in

a first target host of the target network;

installing a first remote agent in the first target host,

the first remote agent being configured to

communicate with the console and a second remote agent; and

executing a second module in the first remote agent,

the second module being configured to

exploit a security vulnerability in

a second target host of the target network.".

12. Claim 15 ***additionally recites*** the limitation that; "The method of claim 14, further comprising

installing a second remote agent in the second target host of the target network,

the second remote agent being configured to

communicate with the first remote agent.".

The teachings of Kargl are directed towards such limitations (i.e., section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.)).

As per claim 17, this claim is the apparatus/system for the method claim 15 above, and is rejected for the same reasons provided for the claim 14 rejection, as such; "The system of claim 15, further comprising

a second remote agent installed in the second target host of the target network,

the second remote agent being configured to

communicate with the first remote agent.".

As per claim 19, this claim is the embodied software for the method claim 15 above, and is rejected for the same reasons provided for the claim 15 rejection, as such; "The computer code of claim 18, further comprising

code for installing a second remote agent in the second target host of the target network,

the second remote agent being configured to

communicate with the first remote agent.".

13. As per claim 20; "A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to

exploit a security vulnerability of

a first target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system

hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.).];

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host [section 2.4-6];

sending a system call to

the first remote agent via a network [section 2.4-6]; and

executing the system call in the first target host using

a proxycall server of the first remote agent to

exploit a security vulnerability of

a second target host [section 2.4-6].".

As per claim 25, this claim is the embodied software for the method claim 20 above, and is rejected for the same reasons provided for the claim 20 rejection, as such; "Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to

exploit a security vulnerability of

a first target host of the target network;

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host;

sending a system call to

the first remote agent via a network; and

executing the system call in the first target host using

a proxycall server of the first remote agent to

exploit a security vulnerability of

a second target host.".

14. As per claim 21; "A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to

exploit a security vulnerability of

a first target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker

compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.).];

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host [section 2.4-6];

executing in the first remote agent

a second module that generates a system call [section 2.4-6]; and

executing the system call in the first target host to

exploit a security vulnerability of

a second target host [section 2.4-6].".

As per claim 26, this claim is the embodied software for the method claim 21 above, and is rejected for the same reasons provided for the claim 21 rejection, as such; "Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to

exploit a security vulnerability of

a first target host of the target network;

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host;

executing in the first remote agent

a second module that generates a system call; and

executing the system call in the first target host to

exploit a security vulnerability of

a second target host.".

15. As per claim 22; "A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to

exploit a security vulnerability of

a first target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker

compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.).];

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host [section 2.4-6];

executing

a second module in the first remote agent that generates a system call [section 2.4-6];

installing a second remote agent in the second target host as a result of

exploiting a security vulnerability of

the second target host [section 2.4-6];

sending the system call generated by the second module to

the second remote agent via a network [section 2.4-6]; and

executing the system call in the second target host using

a proxycall server of

the second remote agent [section 2.4-6].".

As per claim 27, this claim is the embodied software for the method claim 22 above, and is rejected for the same reasons provided for the claim 22 rejection, as such; "Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to

exploit a security vulnerability of

a first target host of the target network;

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host;

executing

a second module in the first remote agent that generates a system call;

installing a second remote agent in the second target host as a result of

exploiting a security vulnerability of

the second target host;

sending the system call generated by the second module to

the second remote agent via a network; and

executing the system call in the second target host using

a proxycall server of

the second remote agent.".

16. As per claim 23; "A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to

exploit a security vulnerability of

a first target host of the target network [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking

using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.).];

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host [section 2.4-6];

installing a second remote agent in the second target host as a result of exploiting a security vulnerability of the second target host [section 2.4-6];

sending a system call to the first remote agent [section 2.4-6];

sending the system call from the first remote agent to the second remote agent [section 2.4-6]; and

executing the system call in the second target host using a proxycall server of the second remote agent [section 2.4-6].".

As per claim 28, this claim is the embodied software for the method claim 23 above, and is rejected for the same reasons provided for the claim 23 rejection, as such; "Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to exploit a security vulnerability of

a first target host of the target network;

installing a first remote agent in the first target host as a result of

exploiting the security vulnerability of

the first target host;

installing a second remote agent in the second target host as a result of

exploiting a security vulnerability of

the second target host;

sending a system call to

the first remote agent;

sending the system call from the first remote agent to

the second remote agent; and

executing the system call in the second target host using

a proxycall server of

the second remote agent.".

17. As per claim 24; "A method for performing penetration testing of a target network, comprising the steps of:

installing a first remote agent in the first target host,

the first remote agent having a proxy server configured to

receive and execute system calls [section 2.4-6, whereas the basic idea behind the distributed denial of service attacks is that the attacker compromises the first network system (i.e., see section 2.4), via performing penetration testing

in order to select possible daemon program/zombie system hosts for the subsequent attack on the targeted server(s). This involves the use of installing remote agents (i.e., inclusive of Root Kits which inherently involve the interception of operating system calls at the kernel level) on the first network system (i.e., an execution engine encompassing routers and firewalls, clearly encompassing proxy servers, and WEB based servers; JAVA capable such that a virtual machine is inherently part of the JAVA functionality, as broadly interpreted by the examiner) and subsequent "zombie" host systems which are clearly capable of receiving, and providing back resulting feedback, for the progress and status of an attack. The initiating attacker also clearly is attacking using a GUI oriented host (i.e., the local agent with a user interface) with associated execution of the attack software and remote agents (i.e., distributed objects, daemons, CGI scripts, etc.).];

executing in the first remote agent

a system call received via a network [section 2.4-6];

installing a second remote agent in the first target host,

the second remote agent having

a proxy server configured to

receive and execute system calls and

a virtual machine configured to

execute scripting language instructions [section 2.4-6]; and

executing in the second remote agent

a scripting language instruction or

a system call, the system call being received via the network [section 2.4-6].".

As per claim 29, this claim is the embodied software for the method claim 24 above, and is rejected for the same reasons provided for the claim 24 rejection, as such; "Computer code for performing penetration testing of a target network, the code comprising code for:

installing a first remote agent in the first target host,

the first remote agent having a proxy server configured to

receive and execute system calls;

executing in the first remote agent

a system call received via a network;

installing a second remote agent in the first target host,

the second remote agent having

a proxy server configured to

receive and execute system calls and

a virtual machine configured to

execute scripting language instructions; and

executing in the second remote agent

a scripting language instruction or

a system call, the system call being received via the network.".

Conclusion

18. Any inquiry concerning this communication or earlier communications from examiner should be directed to Ronald Baum, whose telephone number is (571) 272-3861, and whose unofficial Fax number is (571) 273-3861. The examiner can normally be reached Monday through Thursday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh, can be reached at (571) 272-3795. The Fax number for the organization where this application is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. For more information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Ronald Baum

Patent Examiner